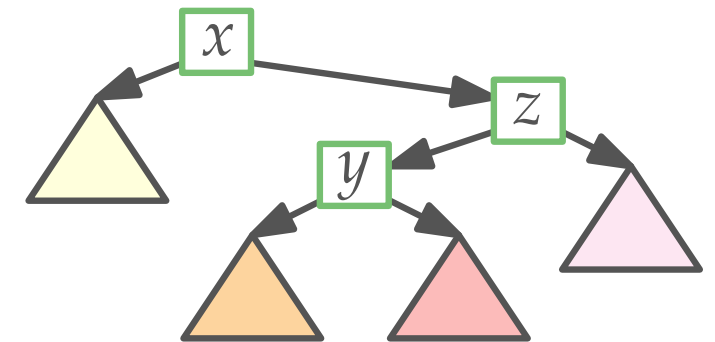
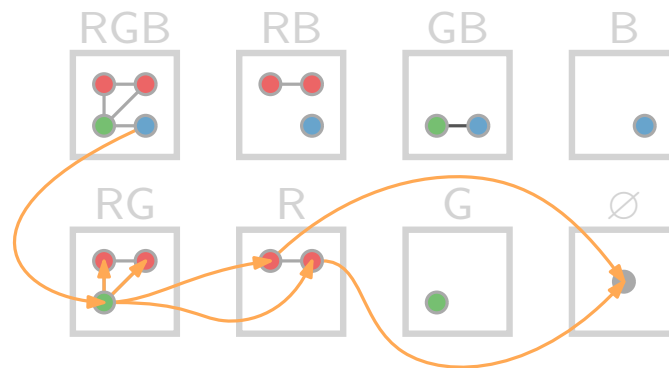
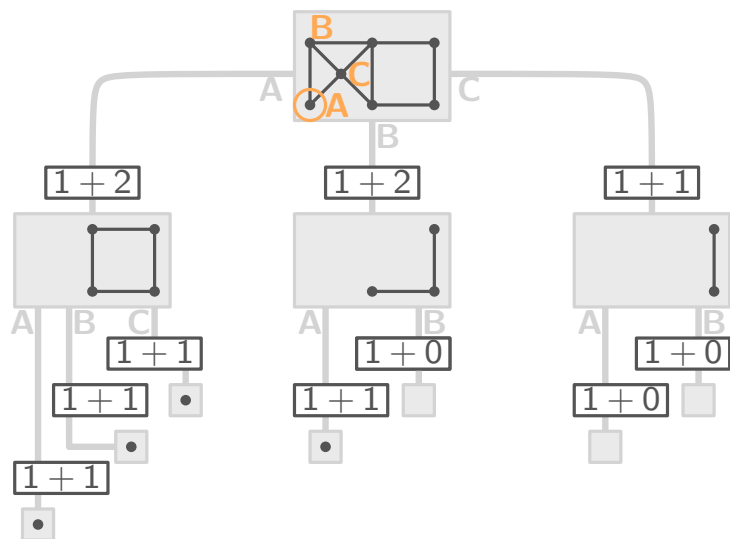


Advanced Algorithms

Introduction

Topics, Course Details, Organizational

Johannes Zink · Alexander Wolff · WS22/23



Advanced Algorithms

The goal of this course is to offer an overview of advanced algorithmic topics.

Advanced Algorithms

The goal of this course is to offer an overview of advanced algorithmic topics.

You have already learned a lot about algorithms

Advanced Algorithms

The goal of this course is to offer an overview of advanced algorithmic topics.

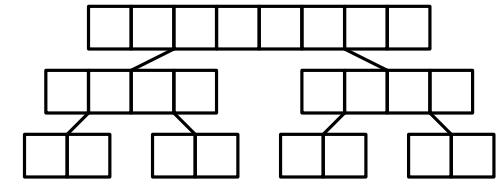
You have already learned a lot about algorithms, but there is much more left...

Advanced Algorithms

The goal of this course is to offer an overview of advanced algorithmic topics.

You have already learned a lot about algorithms, but there is much more left...

- **Types:** incremental, recursive, D&C, greedy, numerical

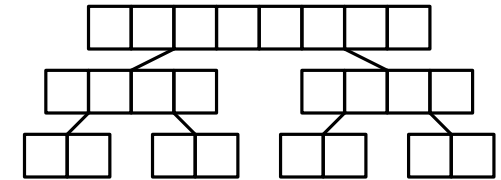


Advanced Algorithms

The goal of this course is to offer an overview of advanced algorithmic topics.

You have already learned a lot about algorithms, but there is much more left...

- **Types:** incremental, recursive, D&C, greedy, numerical, exact, approx., randomized, parallel, distributed, ...

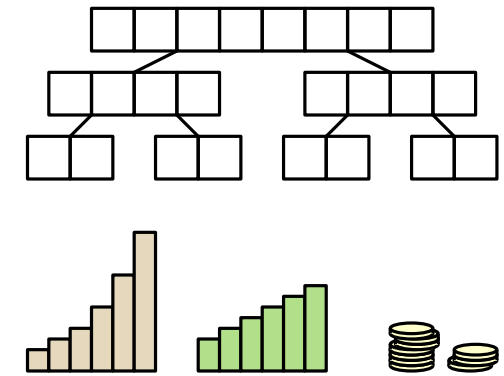


Advanced Algorithms

The goal of this course is to offer an overview of advanced algorithmic topics.

You have already learned a lot about algorithms, but there is much more left...

- **Types:** incremental, recursive, D&C, greedy, numerical, exact, approx., randomized, parallel, distributed, ...
- **Analysis:** correctness, runtime, space usage, amortized

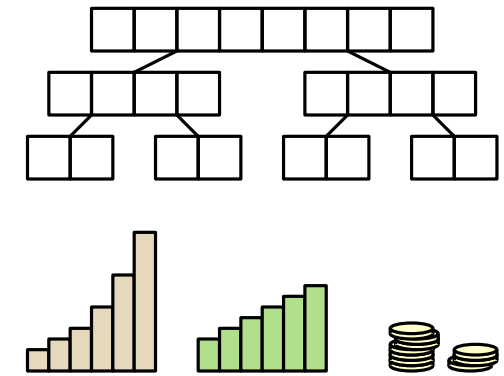


Advanced Algorithms

The goal of this course is to offer an overview of advanced algorithmic topics.

You have already learned a lot about algorithms, but there is much more left...

- **Types:** incremental, recursive, D&C, greedy, numerical, exact, approx., randomized, parallel, distributed, ...
- **Analysis:** correctness, runtime, space usage, amortized, expected, optimality, benchmarking, ...

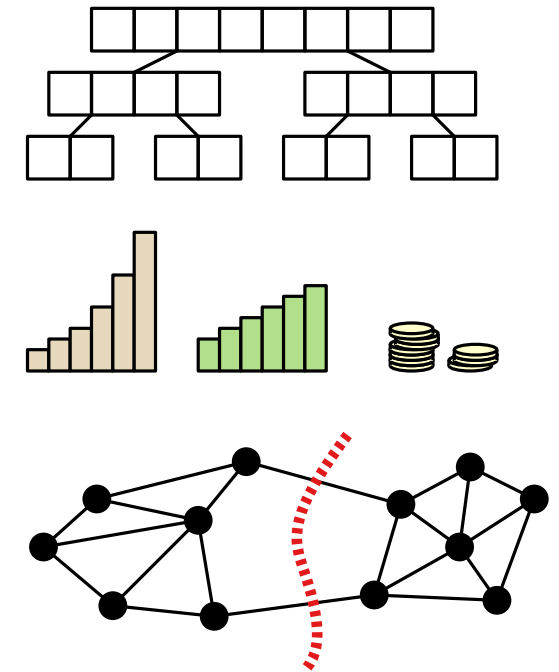


Advanced Algorithms

The goal of this course is to offer an overview of advanced algorithmic topics.

You have already learned a lot about algorithms, but there is much more left...

- **Types:** incremental, recursive, D&C, greedy, numerical, exact, approx., randomized, parallel, distributed, ...
- **Analysis:** correctness, runtime, space usage, amortized, expected, optimality, benchmarking, ...
- **Problems:** combinatorial, graphs, geometric

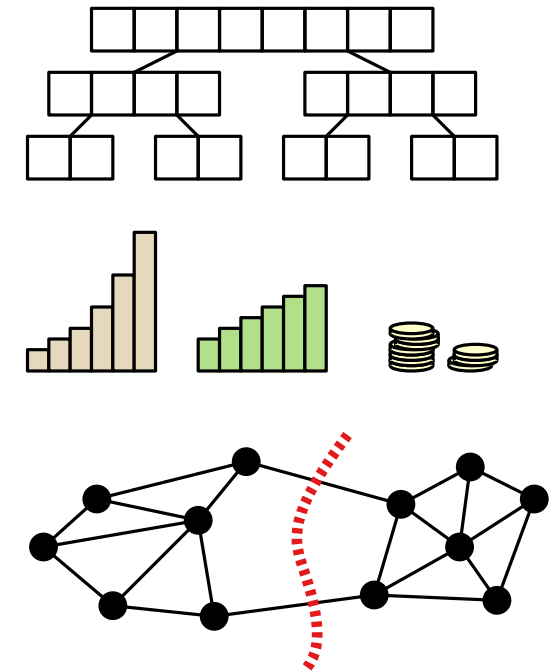


Advanced Algorithms

The goal of this course is to offer an overview of advanced algorithmic topics.

You have already learned a lot about algorithms, but there is much more left...

- **Types:** incremental, recursive, D&C, greedy, numerical, exact, approx., randomized, parallel, distributed, ...
- **Analysis:** correctness, runtime, space usage, amortized, expected, optimality, benchmarking, ...
- **Problems:** combinatorial, graphs, geometric, strings, biological, geographic, ...

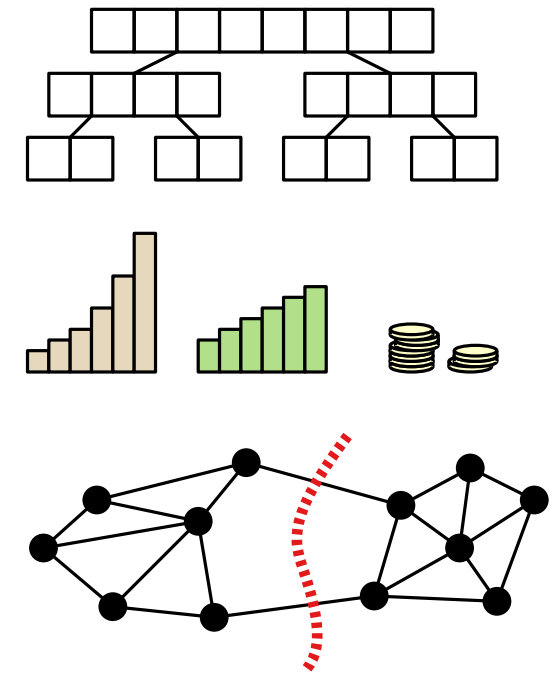


Advanced Algorithms

The goal of this course is to offer an overview of advanced algorithmic topics.

You have already learned a lot about algorithms, but there is much more left...

- **Types:** incremental, recursive, D&C, greedy, numerical, exact, approx., randomized, parallel, distributed, ...
- **Analysis:** correctness, runtime, space usage, amortized, expected, optimality, benchmarking, ...
- **Problems:** combinatorial, graphs, geometric, strings, biological, geographic, ...
- **Data structures:** lists, binary search trees

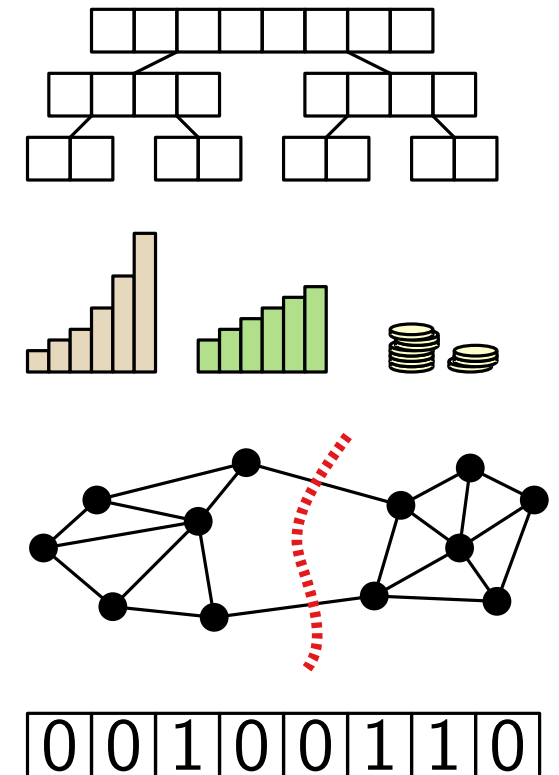


Advanced Algorithms

The goal of this course is to offer an overview of advanced algorithmic topics.

You have already learned a lot about algorithms, but there is much more left...

- **Types:** incremental, recursive, D&C, greedy, numerical, exact, approx., randomized, parallel, distributed, ...
- **Analysis:** correctness, runtime, space usage, amortized, expected, optimality, benchmarking, ...
- **Problems:** combinatorial, graphs, geometric, strings, biological, geographic, ...
- **Data structures:** lists, binary search trees, dictionaries, succinctness, ...

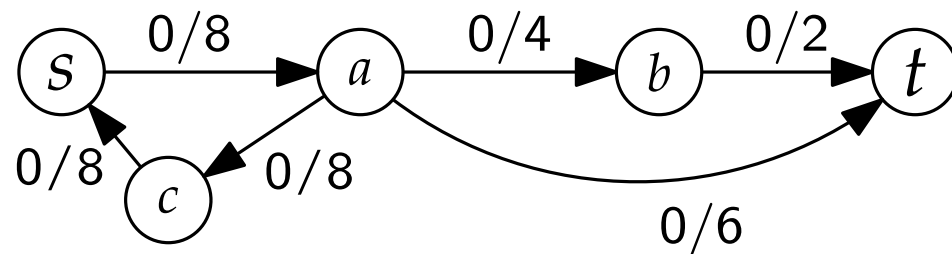


Topics I

- **Better algorithms** for problems you know

Topics I

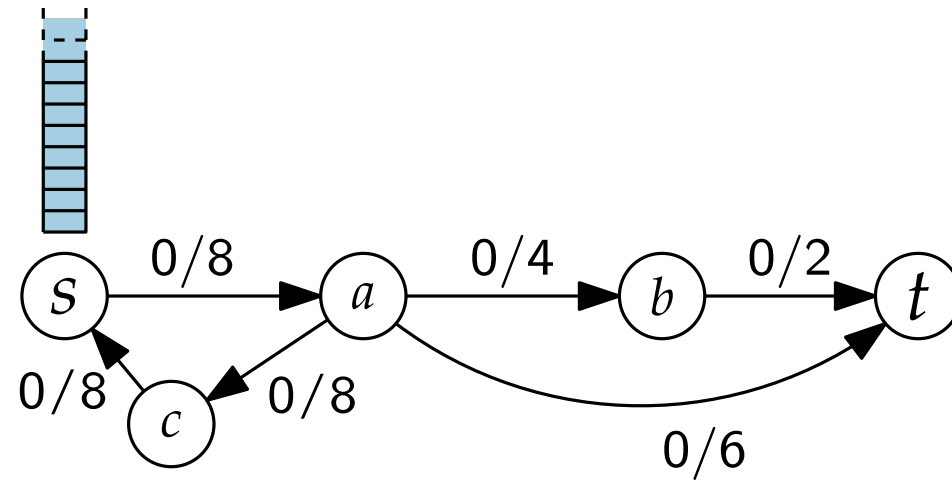
- **Better algorithms** for problems you know
- Maximum flow problem



- Ford–Folkerson algorithm: $\mathcal{O}(|E||f^*|)$
- Edmonds–Karp algorithm: $\mathcal{O}(|V||E|^2)$
- **Push-Relabel** algorithm: $\mathcal{O}(|V|^2|E|)$ (or even better)

Topics I

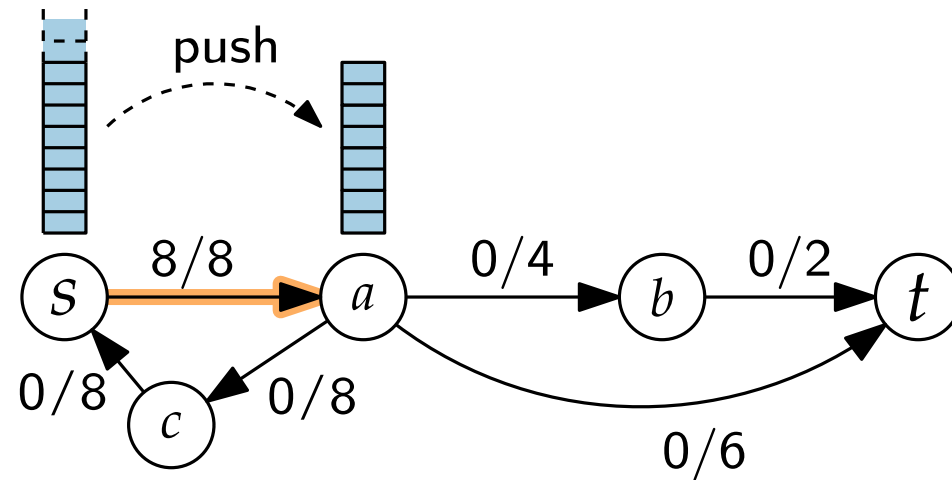
- **Better algorithms** for problems you know
- Maximum flow problem



- Ford–Folkerson algorithm: $\mathcal{O}(|E||f^*|)$
- Edmonds–Karp algorithm: $\mathcal{O}(|V||E|^2)$
- **Push-Relabel** algorithm: $\mathcal{O}(|V|^2|E|)$ (or even better)

Topics I

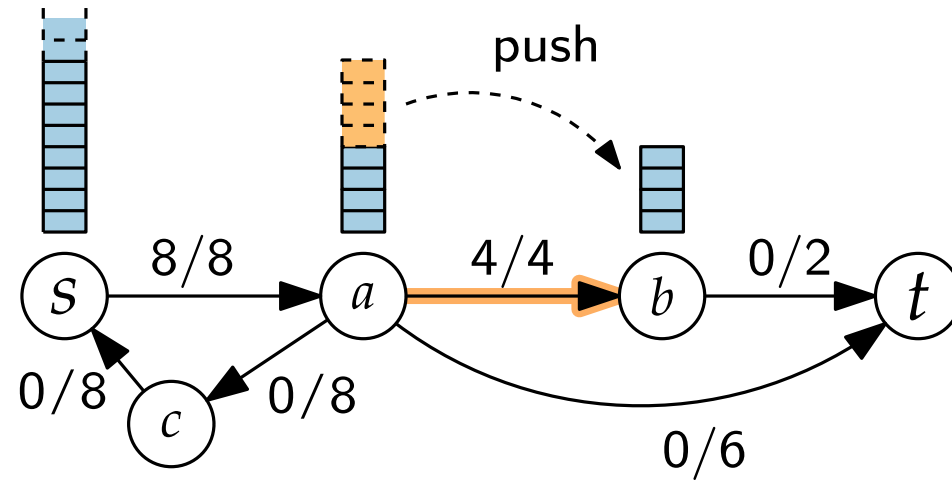
- **Better algorithms** for problems you know
- Maximum flow problem



- Ford–Folkerson algorithm: $\mathcal{O}(|E||f^*|)$
- Edmonds–Karp algorithm: $\mathcal{O}(|V||E|^2)$
- **Push-Relabel** algorithm: $\mathcal{O}(|V|^2|E|)$ (or even better)

Topics I

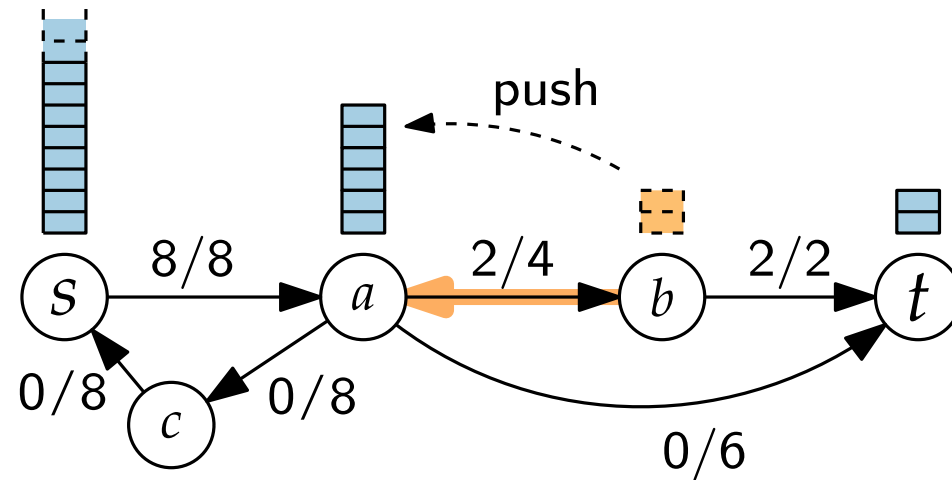
- **Better algorithms** for problems you know
- Maximum flow problem



- Ford–Folkerson algorithm: $\mathcal{O}(|E||f^*|)$
- Edmonds–Karp algorithm: $\mathcal{O}(|V||E|^2)$
- **Push-Relabel** algorithm: $\mathcal{O}(|V|^2|E|)$ (or even better)

Topics I

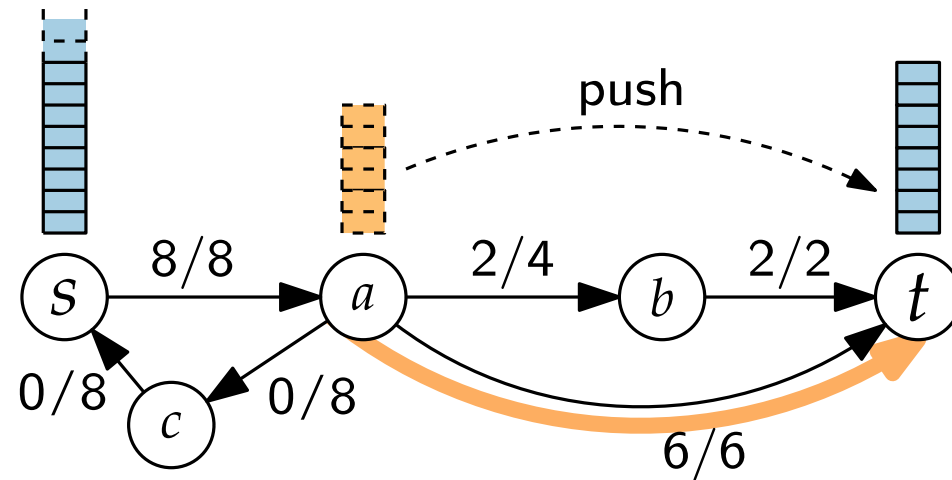
- **Better algorithms** for problems you know
- Maximum flow problem



- Ford–Folkerson algorithm: $\mathcal{O}(|E||f^*|)$
- Edmonds–Karp algorithm: $\mathcal{O}(|V||E|^2)$
- **Push-Relabel** algorithm: $\mathcal{O}(|V|^2|E|)$ (or even better)

Topics I

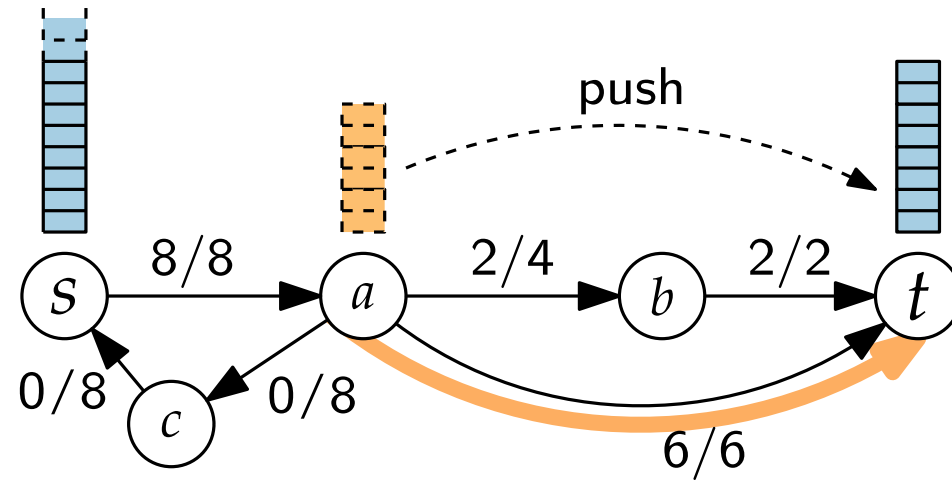
- **Better algorithms** for problems you know
- Maximum flow problem



- Ford–Folkerson algorithm: $\mathcal{O}(|E||f^*|)$
- Edmonds–Karp algorithm: $\mathcal{O}(|V||E|^2)$
- **Push-Relabel** algorithm: $\mathcal{O}(|V|^2|E|)$ (or even better)

Topics I

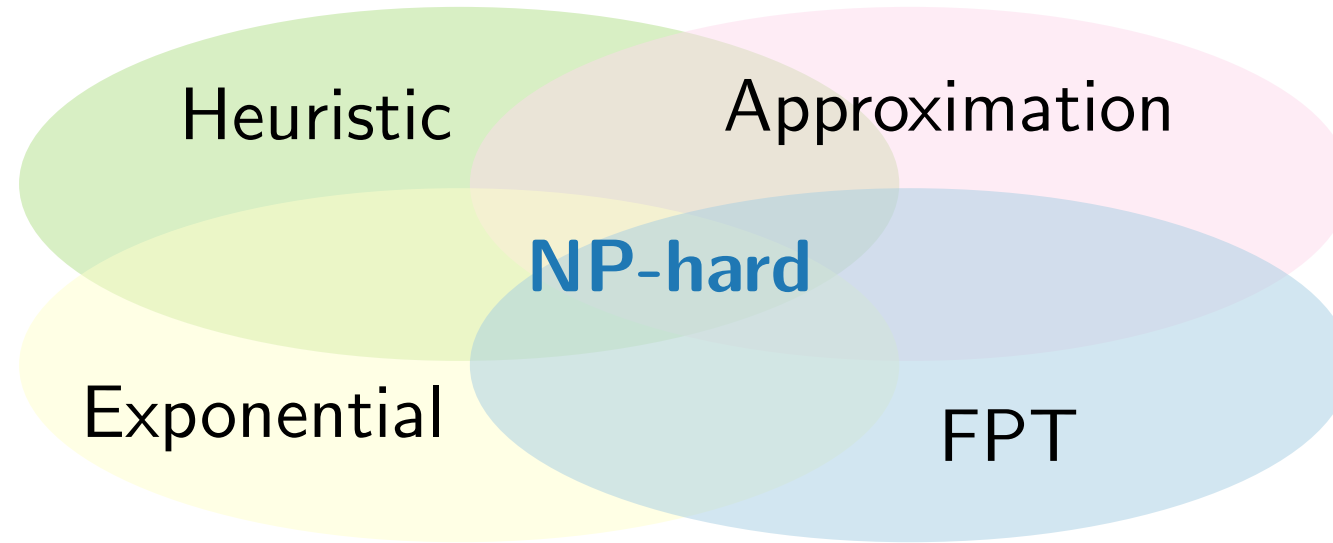
- **Better algorithms** for problems you know
- Maximum flow problem



- Ford–Folkerson algorithm: $\mathcal{O}(|E||f^*|)$
 - Edmonds–Karp algorithm: $\mathcal{O}(|V||E|^2)$
 - **Push-Relabel** algorithm: $\mathcal{O}(|V|^2|E|)$ (or even better)
- Shortest paths in graph with negative edge weights

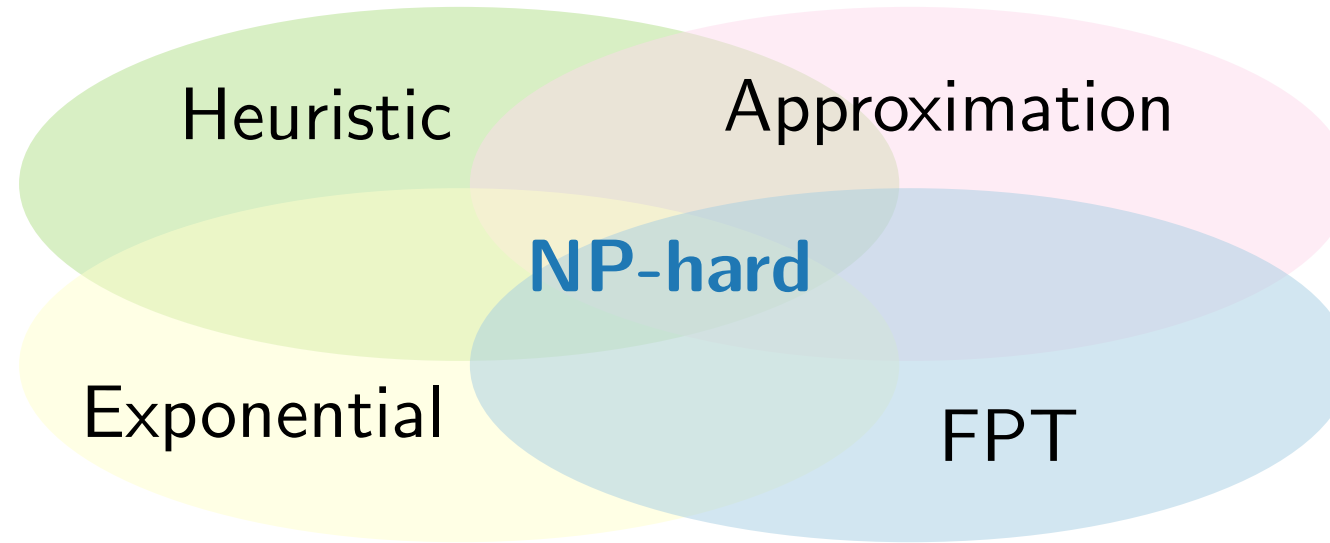
Topics II

- How to deal with **NP-hard problems**



Topics II

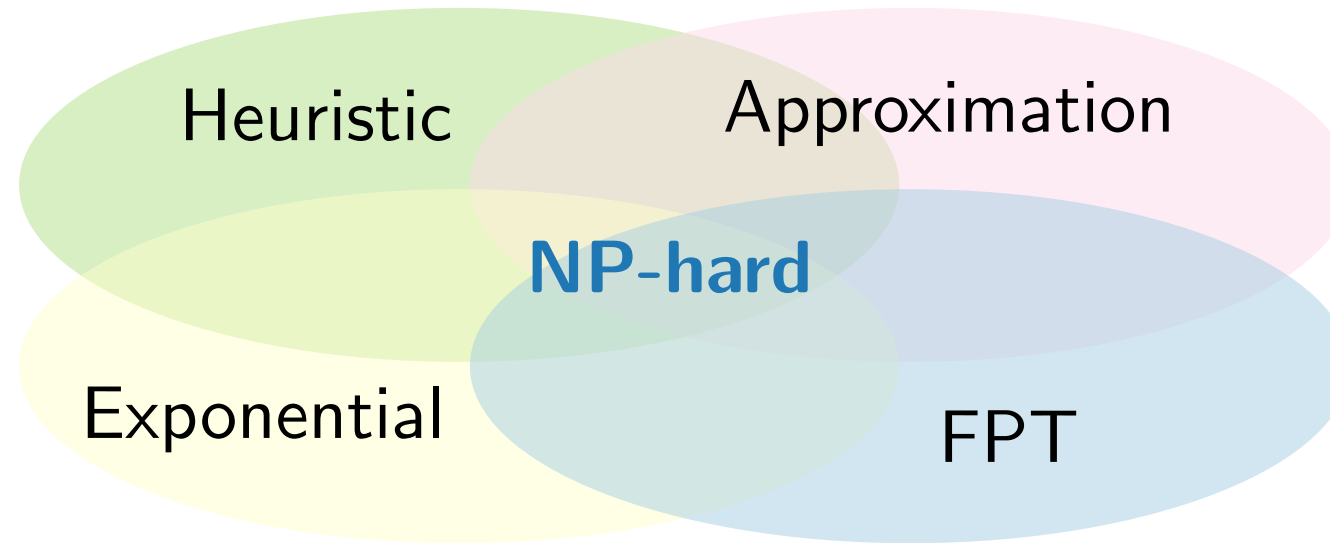
- How to deal with **NP-hard problems**



- Sacrifice quality for speed?
- Can we still compute optimal solutions?

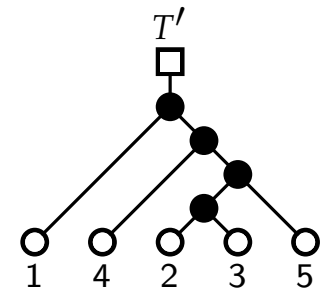
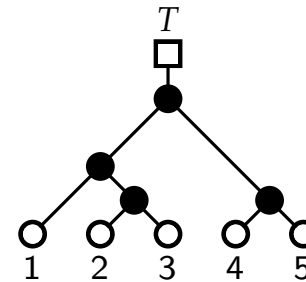
Topics II

■ How to deal with **NP-hard problems**



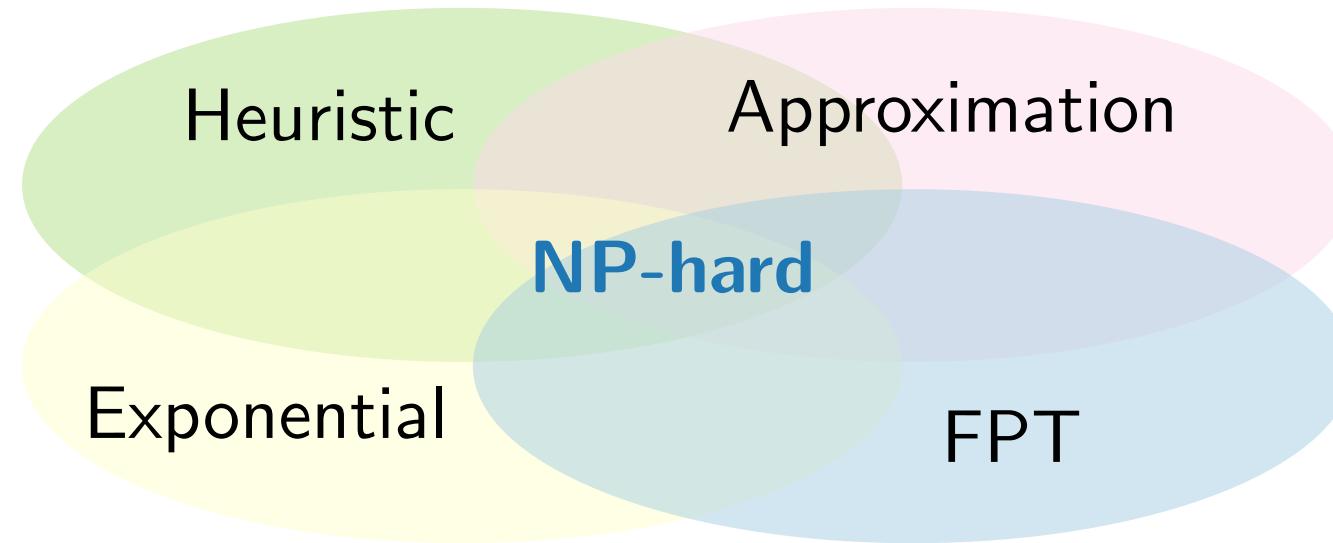
- Sacrifice quality for speed?
- Can we still compute optimal solutions?

- Example problem:
Rearrangement distance of phylogenetic trees



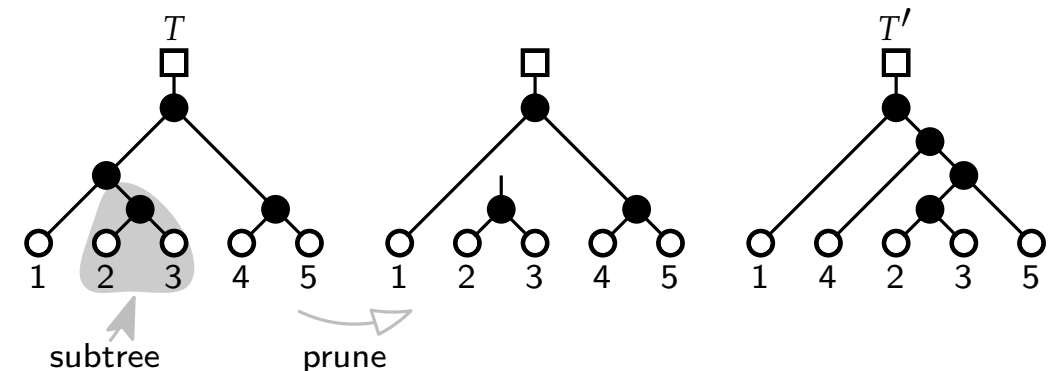
Topics II

- How to deal with **NP-hard problems**



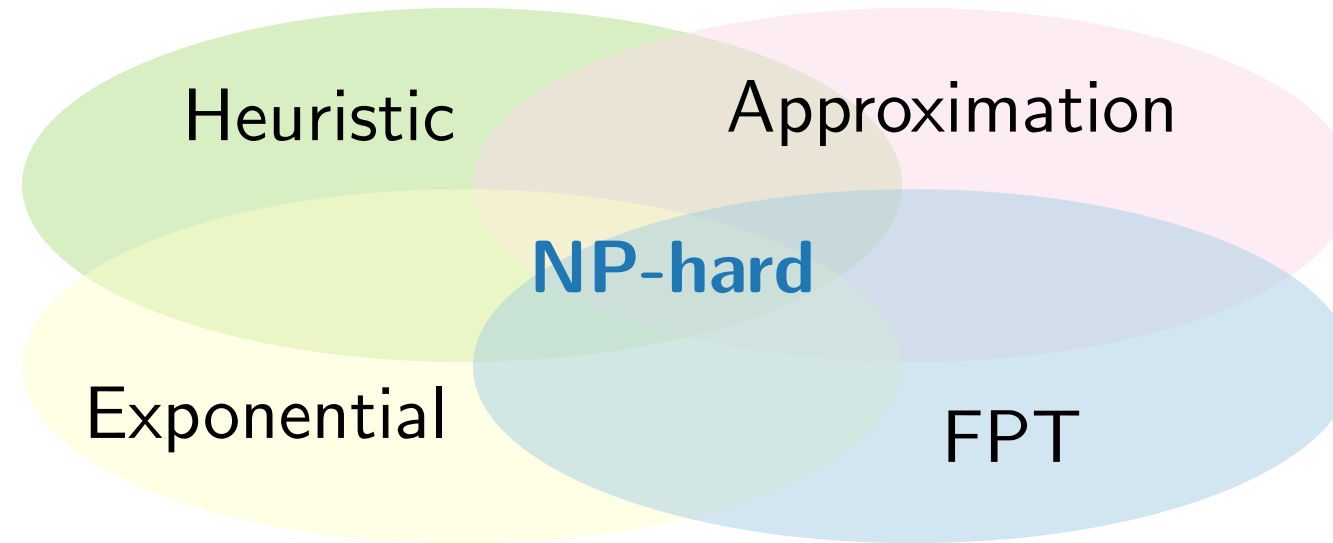
- Sacrifice quality for speed?
- Can we still compute optimal solutions?

- Example problem:
Rearrangement distance of phylogenetic trees



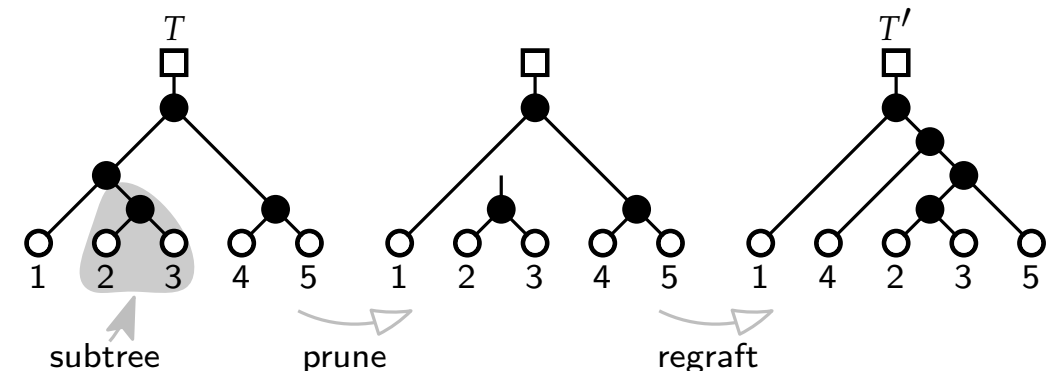
Topics II

■ How to deal with **NP-hard problems**



- Sacrifice quality for speed?
- Can we still compute optimal solutions?

■ Example problem: Rearrangement distance of phylogenetic trees



Topics III

■ Special areas

Randomized algorithms

LONGESTPATH
is NP-hard

but easy on
acyclic digraphs

Topics III

■ Special areas

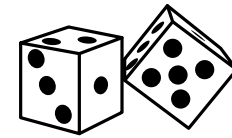
Randomized algorithms

LONGESTPATH
is NP-hard

but easy on
acyclic digraphs



randomly turn given graph
into acyclic digraph



Topics III

■ Special areas

Randomized algorithms

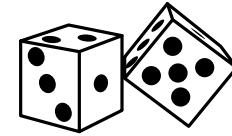
LONGESTPATH
is NP-hard

but easy on
acyclic digraphs



randomly turn given graph
into acyclic digraph

⇒ good idea?



Topics III

■ Special areas

Randomized algorithms

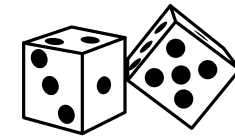
LONGESTPATH
is NP-hard

but easy on
acyclic digraphs

⇒

randomly turn given graph
into acyclic digraph

⇒ good idea?



Also

- Online algorithms
- Approximation algorithms
- Computational geometry
- Working with strings

Topics IV

- (Algorithms for) **Advanced data structures**

Searching for strings

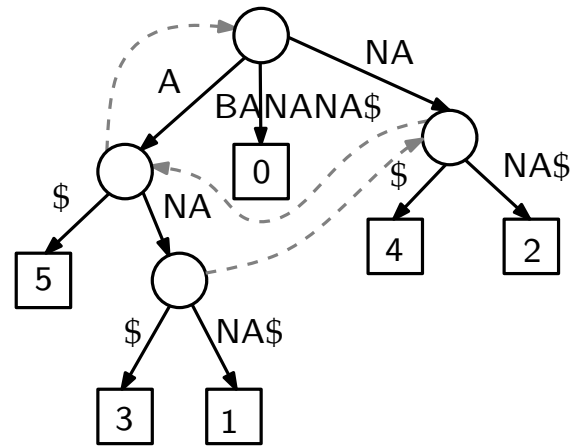
Given text S , how can we efficiently find all occurrences of pattern P ?

Topics IV

■ (Algorithms for) **Advanced data structures**

Searching for strings

Given text S , how can we efficiently find all occurrences of pattern P ?



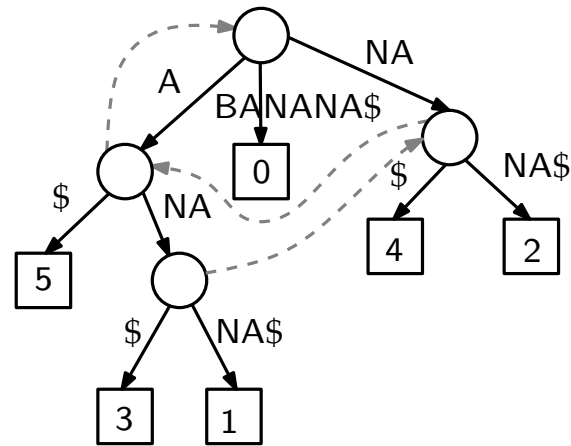
- Suffix trees
- Invest in preprocessing to be faster than full parse

Topics IV

- (Algorithms for) **Advanced data structures**

Searching for strings

Given text S , how can we efficiently find all occurrences of pattern P ?



- **Suffix trees**
- Invest in preprocessing to be faster than full parse

Also

- Succinct data structures
- Splay trees

Lectures

- Johannes Zink (primarily)



- Alexander Wolff (at the beginning)



- Guests: Diana Sieper, Tim Hegemann

Lectures

- Johannes Zink (primarily)



- Alexander Wolff (at the beginning)



- Guests: Diana Sieper, Tim Hegemann
- In-person lectures Wed, 14:15–15:45, ÜR I
- With time for questions and discussions
- Contact also via email: firstname.lastname@uni-wuerzburg.de
- 13–15 lectures
- Old videos from 2020 will be made available on WueCampus

Tutorials

- Oksana Firman



Exercise sheets.

- Weekly exercise sheets, \approx 20 points/sheet
- Released at the lecture day (Wed)
- Submission deadline next lecture (Wed, 14:15)
- Digital submission
 - Recommended to use our LaTeX template
- Submission in teams of two
- Submissions in English (preferred) or German

Tutorials

- Oksana Firman



Exercise sheets.

- Weekly exercise sheets, \approx 20 points/sheet
- Released at the lecture day (Wed)
- Submission deadline next lecture (Wed, 14:15)
- Digital submission
 - Recommended to use our LaTeX template
- Submission in teams of two
- Submissions in English (preferred) or German

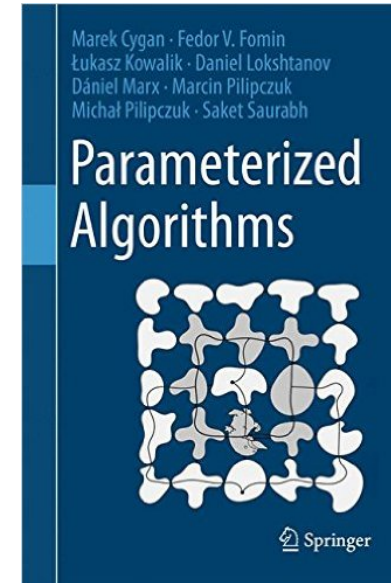
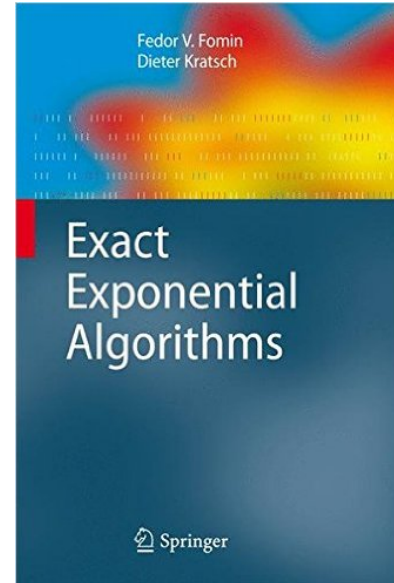
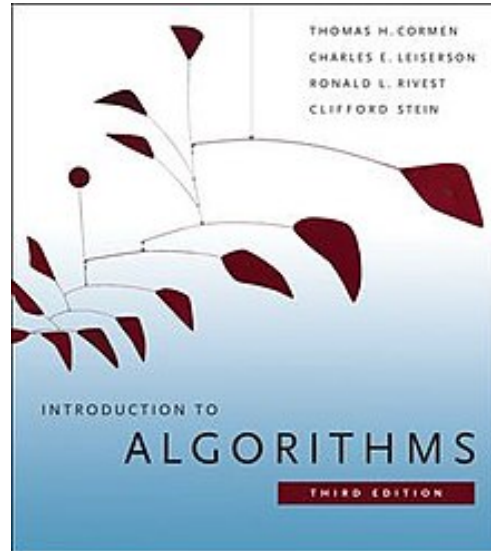
Tutorials.

- In-person Mon 10:15–11:45
- ÜR I
- Discussion of the previous exercise sheet
- Time to work on & ask about the current exercise sheet
- Questions and discussions

Exam

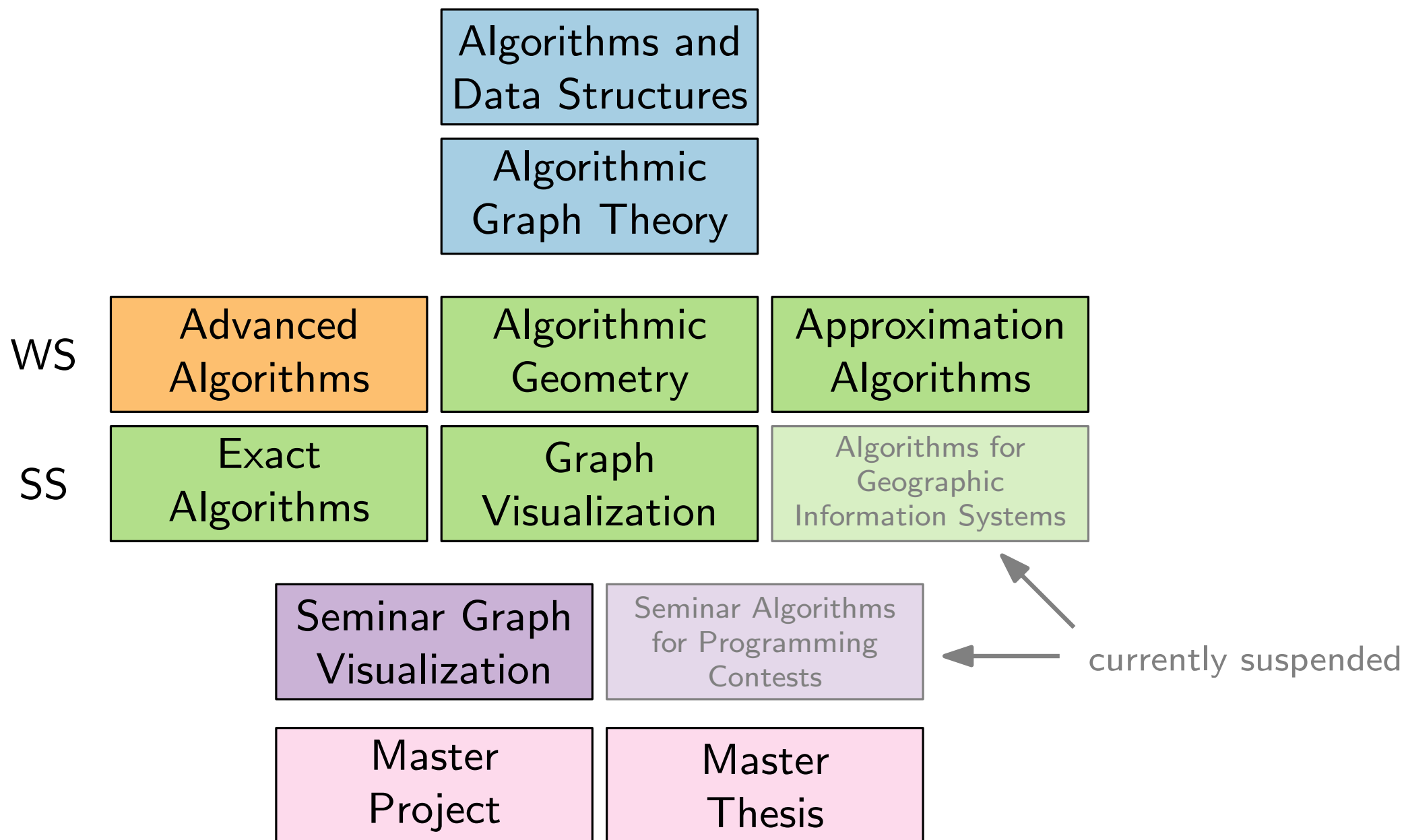
- Oral exam
- \approx 20min
- \geq 50% points on exercises sheets gives one grading level bonus (if passed)
- Date will be announced during the semester
- Don't forget to register in WueStudy:
"Ausgewählte Kapitel der ..."

Literature



- Sources at the end of every lecture
- Links to further interesting stuff

Our Lectures and Seminars



Thanks

Material and slides provided in this lecture have been compiled by many different people. Special thanks to:

Jonathan Klawitter, Boris Klemz, Steven Chaplick,
Thomas van Dijk, Philipp Kindermann, Joachim Spoerhase,
Sabine Storandt, Dorothea Wagner, ...